from tkinter import \*

import random

GAME\_WIDTH = 700

GAME\_HEIGHT = 700

SPEED = 100

SPACE\_SIZE = 50

BODY\_PARTS = 3

SNAKE\_COLOR = "#00FF00"

FOOD\_COLOR = "#FF0000"

BACKGROUND\_COLOR = "#000000"

class Snake:

def \_\_init\_\_(self):

self.body\_size = BODY\_PARTS

self.coordinates = []

self.squares = []

for i in range(0, BODY\_PARTS):

self.coordinates.append([0, 0])

for x, y in self.coordinates:

square = canvas.create\_rectangle(x, y, x + SPACE\_SIZE, y + SPACE\_SIZE, fill=SNAKE\_COLOR, tag="snake")

self.squares.append(square)

class Food:

def \_\_init\_\_(self):

x = random.randint(0, (GAME\_WIDTH / SPACE\_SIZE)-1) \* SPACE\_SIZE

y = random.randint(0, (GAME\_HEIGHT / SPACE\_SIZE) - 1) \* SPACE\_SIZE

self.coordinates = [x, y]

canvas.create\_oval(x, y, x + SPACE\_SIZE, y + SPACE\_SIZE, fill=FOOD\_COLOR, tag="food")

def next\_turn(snake, food):

x, y = snake.coordinates[0]

if direction == "up":

y -= SPACE\_SIZE

elif direction == "down":

y += SPACE\_SIZE

elif direction == "left":

x -= SPACE\_SIZE

elif direction == "right":

x += SPACE\_SIZE

snake.coordinates.insert(0, (x, y))

square = canvas.create\_rectangle(x, y, x + SPACE\_SIZE, y + SPACE\_SIZE, fill=SNAKE\_COLOR)

snake.squares.insert(0, square)

if x == food.coordinates[0] and y == food.coordinates[1]:

global score

score += 1

label.config(text="Score:{}".format(score))

canvas.delete("food")

food = Food()

else:

del snake.coordinates[-1]

canvas.delete(snake.squares[-1])

del snake.squares[-1]

if check\_collisions(snake):

game\_over()

else:

window.after(SPEED, next\_turn, snake, food)

def change\_direction(new\_direction):

global direction

if new\_direction == 'left':

if direction != 'right':

direction = new\_direction

elif new\_direction == 'right':

if direction != 'left':

direction = new\_direction

elif new\_direction == 'up':

if direction != 'down':

direction = new\_direction

elif new\_direction == 'down':

if direction != 'up':

direction = new\_direction

def check\_collisions(snake):

x, y = snake.coordinates[0]

if x < 0 or x >= GAME\_WIDTH:

return True

elif y < 0 or y >= GAME\_HEIGHT:

return True

for body\_part in snake.coordinates[1:]:

if x == body\_part[0] and y == body\_part[1]:

return True

return False

def game\_over():

canvas.delete(ALL)

canvas.create\_text(canvas.winfo\_width()/2, canvas.winfo\_height()/2,

font=('consolas',70), text="GAME OVER", fill="red", tag="gameover")

window = Tk()

window.title("Snake game")

window.resizable(False, False)

score = 0

direction = 'down'

label = Label(window, text="Score:{}".format(score), font=('consolas', 40))

label.pack()

canvas = Canvas(window, bg=BACKGROUND\_COLOR, height=GAME\_HEIGHT, width=GAME\_WIDTH)

canvas.pack()

window.update()

window\_width = window.winfo\_width()

window\_height = window.winfo\_height()

screen\_width = window.winfo\_screenwidth()

screen\_height = window.winfo\_screenheight()

x = int((screen\_width/2) - (window\_width/2))

y = int((screen\_height/2) - (window\_height/2))

window.geometry(f"{window\_width}x{window\_height}+{x}+{y}")

window.bind('<Left>', lambda event: change\_direction('left'))

window.bind('<Right>', lambda event: change\_direction('right'))

window.bind('<Up>', lambda event: change\_direction('up'))

window.bind('<Down>', lambda event: change\_direction('down'))

snake = Snake()

food = Food()

next\_turn(snake, food)

window.mainloop()